

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A document-searching system for searching a document having a hierarchical structure with elements separated by element identifiers, comprising:

a compiling device for generating a query automaton by storing an input query expression, performing parsing, identifying different types of nodes in said element identifiers;

a query automaton storage device for storing the query automaton generated by said compiling device; ~~and~~

a query automaton evaluator for reading out said query automaton from said storage device and storing said automaton, while reading in said document and performing a stream search by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton and outputting the searched node [[[.]]] and

a search result storage means for storing the output of the query automaton evaluator, and for thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.

2. (Currently Amended) The document-searching system according to Claim 1, wherein said query automaton evaluator determines a state transition of a node

under determination ~~at the moment~~ by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node.

3. (Original) The document-searching system according to Claim 1, wherein said compiling device generates a query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon.
4. (Currently Amended) A document-searching method for searching a document having a hierarchical structure with elements separated by element identifiers, comprising the steps of:

generating a query automaton by storing a query expression input by a compiling device, performing parsing, and identifying different types of nodes in said element identifiers;

storing the query automaton generated by said compiling device in a query automaton storage device; and

reading out said query automaton from said query automaton storage device and storing said query automaton, while reading in said document and performing a stream search with a query automaton evaluator by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton[[[.]]] and

storing the output of the query automaton evaluator in a search result storage means for, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.

5. (Original) The document-searching method according to Claim 4, wherein said step of performing a stream search comprises a step of determining a state transition of a node under determination at the moment by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node.
6. (Original) The document-searching method according to Claim 4, wherein said step of generating a query automaton comprises a step of generating a query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon.
7. – 9. (Canceled)
10. (Currently Amended) A computer-readable storage medium storing a computer-executable program for performing a document-searching method for searching a document having a hierarchical structure with elements separated by element identifiers, wherein said program causes a computer to perform the steps of:

functioning as a compiling device for generating a query automaton by storing an input query expression, performing parsing, and identifying different types of nodes in said element identifiers;

storing ~~the~~ a query automaton generated by said compiling device in a query automaton storage device; ~~and~~

functioning as a query automaton evaluator for reading out said query automaton from said storage device and storing said query automaton, while reading in said

document and performing a stream search by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton [[[.]]] and

storing the output of the query automaton evaluator in a search result storage means, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.

11. (Original) The storage medium according to Claim 10, wherein said performance of a stream search determines a state transition of a node under determination at the moment by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node, and wherein said query automaton is generated as a query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon.

12. – 13 (Canceled)

14. (Currently Amended) A compiling method for generating a query automaton for performing a document search, comprising the steps of:  
generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, and storing a plurality of states of said a backward node in correspondence with said backward node into a query automaton storage device; ~~and~~

generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, and a reached state in correspondence with each other in said storage device [[[.]]] and

storing the output of the query automaton evaluator in a search result storage means, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.

15. (Original) The compiling method according to Claim 14, wherein said compiling method comprises a step of identifying said backward node as a left node or a lower node according to a type of said element identifier, and wherein said plurality of states are states of said left node and said lower node.

16. – 17. (Canceled)

18. (Currently Amended) A computer-readable storage medium storing a program for causing a computer to perform a compiling method for generating a query automaton for performing a document search, wherein said program causes a computer to perform the steps of:

generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in term of search, and storing the plurality of states of said a backward node in correspondence with said backward node into a storage device; and

generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, and a reached state in correspondence with each other in said storage device [[[.]]] and

storing the output of the query automaton evaluator in a search result storage means, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node

19. (Original) The storage medium according to Claim 18, wherein said program comprises a step of causing a computer to identify said backward node as a left node or a lower node according to a type of said element identifier, and wherein said plurality of states are states of said left node and said lower node.

20. (Currently Amended) A document-searching system for searching a document having a hierarchical structure with elements separated by element identifiers, comprising:

a compiling device for generating a two-state input automaton for enabling a state transition by storing an input query expression, performing parsing, and reading at least two states assigned to different types of nodes in said element identifiers;

[[[a]]] an automaton query storage device for storing said two-state input automaton; and

an automaton-evaluating device for enabling a state transition by reading out two-state input automaton from said storage device and storing said automaton, while reading in said document and identifying said two states node [[[.]]] and

a search result storage means for storing the output of the query automaton evaluator, and for thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.

21. (Original) The document-searching system according to Claim 20, wherein said two states are states of a left node and a lower node of a tree structure generated in correspondence with an identified element identifier, and wherein said two-state input automaton uses three states of said automaton-evaluating device.

22. (Currently Amended) A query automaton evaluator for evaluating a query automaton for searching a document having a hierarchical structure with elements separated by element identifiers, comprising:

means for reading out a query automaton from a query automaton storage device that enables a plurality of inputs generated by a compiling device to be determined at a time and storing the query automaton;

means for identifying a plurality of different types of inputs of said element identifiers included in said document;~~and~~

means for assigning a state transition among three states including a search state by using said identified input and a plurality of inputs registered in said query automaton node [[[.]]] and

a search result storage means for storing the output of the query automaton evaluator, and for thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.